GLAST MOC Secure File Transfer Product Selection

30 June 2004
Marilyn Mix
marilyn.mix@omitron.com

Issue

The GLAST MOC development team needed to select a file transfer mechanism to be used from the ground stations to the MOC and from the MOC to the science facilities. The telemetry files and thus the Level 0 data sets for GLAST are going to be so large that a different file transfer technology than Secure Shell (ssh) is going to have to be used. Primarily, the ability to resume the file transfer after a network or computer failure is necessary. Newer FTP clients/servers do have a resume capability, but they are not secure.

Trade Study

The MOC development team performed a trade study of file transfer products. The evaluation criteria are:

1. Reliability (guaranteed delivery, data integrity, e.g., checksum or message digest)
2. Security (encrypted, user authentication, password protected)
3. Ability to resume transmission for failed/interrupted transfers of large files by restarting the transfer from a checkpoint instead of from the beginning of file.
4. Cross-platform support for Solaris, Linux, and Windows platforms
5. Ease of setup and maintenance
6. Technical Support (response time, helpfulness)
7. Cost

The MOC team evaluated several commercial products as well as a few no-cost packages. The evaluation is summarized in the **Detailed Evaluation** section.

Recommended Solution

The MOC development team recommends using Softlink's FASTCopy product for the secure transfer of large files to and from the MOC. A license for FASTCopy is required for each workstation. Pricing is based on the number of processors in a workstation. Softlink doesn't seem to have site license pricing, but indicated that there would be a discount for the organization (NASA). For a single processor workstation the price is on the order of $1000 and includes maintenance (tech support and updates). This is a bargain considering it meets or exceeds our criteria "out of the box". Swift spent several months of effort developing its file transfer mechanism.

The MOC team acquired an evaluation copy of FASTCopy. It works as advertised for the features the MOC requires. It is less expensive than comparable commercial software. FASTCopy is used as the file transfer protocol for the Standard Autonomous File Server

(SAFS) used by many NASA Ground Network sites to provide automated file transfer. The file transfer times for uncompressed data is comparable to FTP, and typically improve when on-the-fly compression is used. DTS was the only no-cost software that met the minimum MOC requirements. It was developed by LHEA which is one of the sites we will be transfering files to. It uses email and SSH file transfer. It has been used for more than a decade in several missions (but not in a MOC).

FASTCopy is better than DTS for use in the MOC for the following reasons:

1. FASTCopy does not use SSH. The MOC doesn't want external users to have access to the SSH port on our systems. There is no adequate way within open ssh to prevent an external user from copying files from where they are not supposed to. This is a key hole in our Swift design.

2. FASTCopy does not use system accounts for external access. FASTCopy allows the definition of proxy accounts with passwords that it maps to system accounts. Thus, the actual account passwords can be changed regularly and it will be transparent to the user. We don't wan't to give external users system accounts to a MOC computer. As of this writing, nearly all major distributions of ssh servers require system accounts. SSH distributions that plug these holes probably cost more that FASTCopy alone. Although the risk can be plugged to some extent with null shells, it is still a risk for external users to have system accounts and passwords. DTS requires system accounts for file transfer. They will have null shells so they can't be used to log into the system.

3. FASTCopy has additional directory and file restrictions that will allow us to restrict access to certain directories. It limits read, write, and execute privileges on a directory or file-by-file basis. SSH's SCP and SFTP file access is not restrictable. An external user has all the read and write privileges granted to them of ANY file on the system that their system account has read and write privileges over. This is another key hole in our Swift design. DTS always does a file pull only from the directories that the originator specifies and into a specified local directory. However, since the SSH port is open and the account information is known through the dts-sitelist, it is possible for an SFTP to be issued outside of DTS.

4. FASTCopy is easier to install, maintain, and troubleshoot. The DTS email interface adds yet another layer of complexity that the MOC is not willing to accept. We will not have sendmail access to the MOC. It is possible to have yet another piece of client software (fetchmail) poll external mail servers to retrieve mail. Also, the hostname, account name, and passwords are distributed to participating sites whenever the password is changed (every 30 days). This is a maintenance activity that the MOC would prefer to avoid.

Detailed Evaluation

Vendor	Softlink	LHEA	Tumbleweed	Proginet	Sterling	GNU
Product	**FASTCopy**	**DTS**	**Secure Transport**	**CyberFusion**	**Connect Direct FTP+**	**rsync**
Reliable	yes	no guarantee mail delivery	yes	yes		yes* retry not automatic
Secure	SSL	yes (sftp)	yes	yes	yes	ssh
Resume	yes	yes	yes	yes	yes	yes
Cross-platform	yes	yes * Perl	yes	yes	yes	yes * Cygwin
Easy setup and maintenance	yes (vendor help needed to select the necessary options)	moderate (Extra accounts, fetchmail, periodic site file distribution)	no demo	no demo	client/ moderate server/?	yes
Technical Support	Excellent. Prompt and knowledgable. phone, web, email. Excellent documentation	yes. Developers are local. Source code available.	-	late response to request for pricing	web	little
Cost	$1000/workstati on	none	???? server/ $300 client	$6500/worksta tion	$??? server/ free client	none* development cost to add automatic retry

Blank or ??? means unknown

Softlink FASTCopy

http://www.softlink.com
Used with SAFS at Wallops; contact Susan.K.Semancik.1@gsfc.nasa.gov
http://www.wff.nasa.gov/~websafs/index.html.
Other customers include JPL, and NOAA.
See Dustin's technical evaluation notes – overall positive

FASTCopy is their proprietary file transfer protocol. We would need a copy on each machine we are transferring the large files between.

Sales and Tech Support are excellent. Primarily via email. However, they responded promptly and persistently (after noting an email problem, they telephoned me.) Prompt, conscientious, knowledgeable sales and tech support. Discussion of right architecture for our application (individual licenses versus server) Didn't push expensive solution.
Offered help during evaluation period
Easier/cheaper since it's within an organization (NASA)
FASTCopy is available Red Hat Enterprise 3.0 Linux and Red Hat 7.x Linux. (7.x works for Fedora)

<u>Major Features:</u>

- Guaranteed file delivery including:
    - Recovery from the point of failure within a block in the file.
    - Manual and automatic recovery.
    - Hold and Resume capability for very large transfer operations.
- Scheduler for transfer operations.
- Central monitoring and control for problem tracking by operators.
- Enhanced integrity features
- Complete file transfer security and administration including:
    - Access Control features: rule-base screening, eliminating the need for usernames or passwords in scripts, resource protection and logical usernames use.
    - Untrusted network (i.e. Internet) features including encryption and authentication. SSL
    - Untrusted local users features.
- Post-Transfer Processing (lets you specify what process / programs will be executed at the end of the transfer on both nodes) eliminating the need to use "file watchers" to kick-off tasks at the end of transfers.
- Windows and Unix
- Callable Application Interface (API), designed to integrate into other applications and products. C, C++, Java (wrapper). No mention of Perl, so we'd use a system call to fcopy.
- Command line or GUI
- Central on-line monitoring over all network wide transfer operations:
- Provides immediate problem tracking and handling to minimize application downtime.
- Open report and alert system that can send data to any external monitoring system.
- Keeps historical information for post-mortem case studies and capacity planning.

Obtained evaluation copy and verified these features in the limited time we spent on it:

- Proxy usernames and passwords. Usernames and passwords can be completely different than system usernames and passwords. So distribution of usernames and passwords for FASTCopy will not compromise the rest of the system.
- Full wildcard “scp like” source and destination filename references for single and multiple files at once.
- File transfer resume after failure.

- Batch mode. i.e. a program can spawn off the transfer and continue with other activities.
- Batch mode monitoring and logging.
- Restriction of access to FASTCopy process by IP address on client AND server side.
- Permissions of file access and push and pull transfer can be restricted at system, directory, and file level. This is the huge problem with ssh.

DTS

Laboratory for High Energy Physics
http://heasarc.gsfc.nasa.gov/dts/

The DTS uses sendmail only as a means of transporting a file list from site to site. fetchmail may be used instead where security prohibits sendmail. Actual data transfers are accomplished via SFTP.

DTS has a capability to checkpoint/resume in case of transfer failures. The file is split in to pieces and then reassembled after transfer. If any piece fails, it is retransmitted. Once all the pieces arrive safely, they are reassembled.

The DTS is a single perl script. DTS developers are local. The source code is available so we could modify it as needed.

DTS has two, asynchronous 'modes' (send and get). It works as follows (at its most simple, it has other 'features' like zipping/cleaning/recovering/etc):

Site 1 runs the DTS in 'send' mode whereby the script takes as input an ascii file list, or a typeglob (similar to unix 'find') of local files it wants Site 2 to receive.

It 'analyzes' (determines checksums and file sizes for each file) the list, and copies all files to a 'staging area' (an FTP staging chrooted to '/'). It sends an email message with that information to the receiving site. This message goes to an email-only account (e.g. 'dtsmail') at the receiving site that has a null shell and can only be accessed with ssh/scp on the local machine (i.e. the known_hosts file has only fully rendered machine names, e.g. 'heaswift.gsfc.nasa.gov' not just 'heaswift').

The DTS is then run at Site 2 in 'get' mode. The script, 'scp' copies the mailbox of the dtsmail account to a log directory, then processes each message in the mailbox (it throws out spam and other messages that don't fit the dts 'pattern'). Reading the mail message, the script determines the sending site, and looks up that site's machine name, and username/passwd in an encrypted local file, runs SFTP to that site, downloads only the files that are in the send message, checks that they are all the same files (it rejects any file that once retrieved does not have the correct name/size/checksum as listed in the email). It then sends an 'acknowledgement' to the sending site, and that site can use that acknowledgement to automatically 'clean' the staging directory, completing the transfer 'circle'.

The dtsftp and dtsmail accounts have no shells, so even if their passwds became known, no one can use them to access the system (also the FTP staging area doesn't have write privilege except for local dts 'group' members).

The mail messages sent look like this:

```
Date: Thu, 13 Jul 2000 10:03:00 +0100 (BST)
Message-Id: <22714.200007130903@site1.gov>
To: dtsmail@site2.gov
Subject: DTS SEND dts000713_100250SSITE1SITE2

--- SEND initiated from SITE1 to SITE2
SITE SITE1
MODE SEND
TYPE NONE
COMPRESS NONE
DIR ./dts000713_100250SSITE1SITE2
SEQNUM dts000713_100250SSITE1SITE2
DATE 963478979.928187
Thursday July 13 09:02:59 GMT 2000
3 FILES
10010025 BYTES
filename  size (b)     checksum
t1        1            68b329da9893e34099c7d8ad5cb9c940
t10000k   10240000     b690d2a4639f0fc4437fca9fd915a05f
t1000k    1024000      cada2daa26f68ee21465b5e9af90332f

END FILE LIST
--- SEND completed to dtsmail@site2.gov Thursday July 13 09:03:00 GMT 2000
```

Good features are:

- including checksums,
- limiting the transfer to the notification list,
- using local host management,
- and keeping password storage encrypted.

The HEASARC views the DTS as 'open source'-like software. GLAST can have the code and modify it to use something other than sendmail. HEASARC simply won't support that new version.

Current objections to DTS from GLAST MOC standpoint:

1. sendmail is trouble to administer from a security standpoint. Sendmail typically has a lot of security vulnerabilities if not patched often. Keeping port 25 open is a security issue. Setting up fetchmail and the email server is extra setup and another source of troubleshooting that the FOT must deal with.
2. Incoming sendmail may not be allowed on the Restricted IOnet. So the fetchmail option must be used, including setting up an external mail server.
3. email notifications are not guaranteed to be delivered. If the notification is not delivered, then the file transfer is not initiated.
4. As currently designed, the openssh program does not restrict scp users to scp. They also have ssh AND shell capabilities. The commercial ssh program does have this restriction, but the MOC is not using the commercial version of ssh.
5. maintenance of passwords – passwords should change periodically. The dts-sitelist will need to be updated periodically. This is a maintenance activity we'd rather avoid. FASTCopy's proxy user capability shield the senders from system account password changes.

Summary of GSSC objections to FASTCopy and possible amelioration.

1. When they evaluated it, FASTCopy was not available for Fedora. The Red Hat 7.x Linux version runs under Fedora too. Fedora and Red Hat Linux versions are likely to remain compatible. SoftLink is very responsive in updating the software for newer Linux versions. They created a version of Red Hat Enterprise 3.0 becase we requested it.
2. FASTCopy will push files until the file system is full, then it retries the transfer after a delay (hoping someone will increase the free disk space in the interim). The GSSC file server should be designed with sufficient disk space.
3. Post transfer processing. Sender must indicate a script to be called. This is similar to the DTS file type indication which maps to a script. The script can be a link, so that the actual script name can change without impacting the sender. FASTCopy can be set up so that only certain users can execute certain scripts and no other scripts can be executed. You can't transfer a script and then execute it.
4. Receipt path issue– not entirely true. FASTCopy can be configured on the receiver's side so that the desired destination directory is "root" directory of the transfer account. The sender does not need to know the directory structure.
5. Logging and round trip data verification - Additional coding may be needed for the desired amount and kind of data accountability. FASTCopy does provide reports and remote monitoring of files transfers which may support the desired data accountability. Batch mode monitoring and logging was successfully tested by the MOC. Explicit acknowledgement messages may not be necessary due to FASTCopy's reliable protocol.
6. Long term usage. FASTCopy is a commercial product. What if SoftLink goes out of business or stops supporting FASTCopy for future platforms? The MOC is based on COTS products where feasible. Other NASA facilities use FASTCopy for mission critical systems (SAFS). Other customers are JPL and NOAA.
7. Cost. The money saved in testing and troubleshooting should outweigh the dollar cost.

8. GSSC already has a DTS infrastructure in place. Using FASTCopy require some additional effort.

Tumbleweed Communications

http://www.tumbleweed.com/en/products/secure_transport.html
Available for Solaris, Red Hat Linux, Windows
mid-file transfer recovery, checkpoint/restart
multi-gigabyte file size
automation/scripting
encrypts, authenticates
guaranteed delivery
Command line and API for applications (SDK)
Users: banks, government

Client $300 according to Valicert, a division of Tumbleweed
No demo version available from the web site. Limited response from Sales, so no detailed pricing.

Proginet Corporation CyberFusion

http://www.proginet.com/ftpfiletransfer.asp

Security, guaranteed delivery, checkpoint/restart,
Free client software (browser plug-in?)
Minimal info on web site

Response from info query was not prompt:
They sent a brochure. Looks like it has the features we need.
Spoke with an engineer:
Does everything FASTCopy does (they claim) plus web based tool for monitoring transfers.
Cost $6500 per node.
Did not attempt to get an evaluation copy because the cost/benefit ratio was not as good as FASTCopy

Sterling Commerce Connect:Direct FTP+

http://www.sterlingcommerce.com/solutions/connectdirect/ftp/index.html

Works behind existing FTP scripts to replace standard FTP operation with the managed, secure data delivery features of Connect:Direct. It looks like FTP, plus:

- Data security (encryption)
- Data integrity (no lost or corrupted data)
- Checkpoint/restart
- Data compression
- The ability to manage connections and audit all file transfers consistently at the Connect:Direct server.

Downloaded demo code and docs. Attempted to use client with their server. It failed. No response to emailed request for help. Not impressed with sales or tech support.

Java J2SE 1.3
Solaris 8, Linux/Intel, Windows
Client is free. Price for server is ?
Price for security module is ?
Must include Connect:Direct Server, probably at the MOC with clients at the ground stations and GSSC/IOCs that can push or pull.
Not point to point like FASTCopy
Command line interface similar to ftp. You can invoke cdftp with a script that contains the put or get commands.

rsync –

GNU open source
Intended as a replacement for rcp
Designed to send only the parts of the file that are different, server reassembles file from the pieces. Uses exchange of file checksums to find the differences.
By default, it uses ssh.
rsync daemon (rsyncd) handles incoming rsync commands, ala ftpd
The authentication protocol used in rsync is a 128 bit MD4 based challenge response system.
Use ssh as the transport if you want encryption.
Compression: zlib
Platforms: Unix, Linux, Cygwin
Used in local mode: It only copies changes from source to destination. It does not delete files that are in the destination but not in the source.
Evaluation: Does not meet all criteria for file transfer. However, it is great for local file backups.